



# PREMIUM AUDIT

NAME OF PROJECT	Meta Ninja Games
TOKEN BEP20 ADDRESS	0xeEC47a8FcC94c3cA26f7473CDfDc3FF2d8B05bB3
WEBSITE / TELEGRAM	<a href="https://www.metaninja.games/">https://www.metaninja.games/</a>



**DISCLAIMER: PLEASE READ FULL AUDIT**



# IMPORTANT DISCLAIMER

---

I.

## **NOT RESPONSIBLE**

Our audit report is not responsible for anything if the project audited turn into a scam.

II.

## **NOT GUARANTEE**

Analytix Audit in no way guarantees that a project will not remove liquidity, sell off team tokens, or exit scam.

III.

## **INFORMATION**

Analytix Audit do a research and provides public information about the project in an easy-to-understand format for the common person.

IV.

## **AUDIT AGREE**

Agreeing to an audit in no way guarantees that a team will not remove all liquidity ("Rug Pull"), remove liquidity slowly, sell off tokens, quit the project, or completely exit scam

V.

## **ENCOURAGE INVESTMENT**

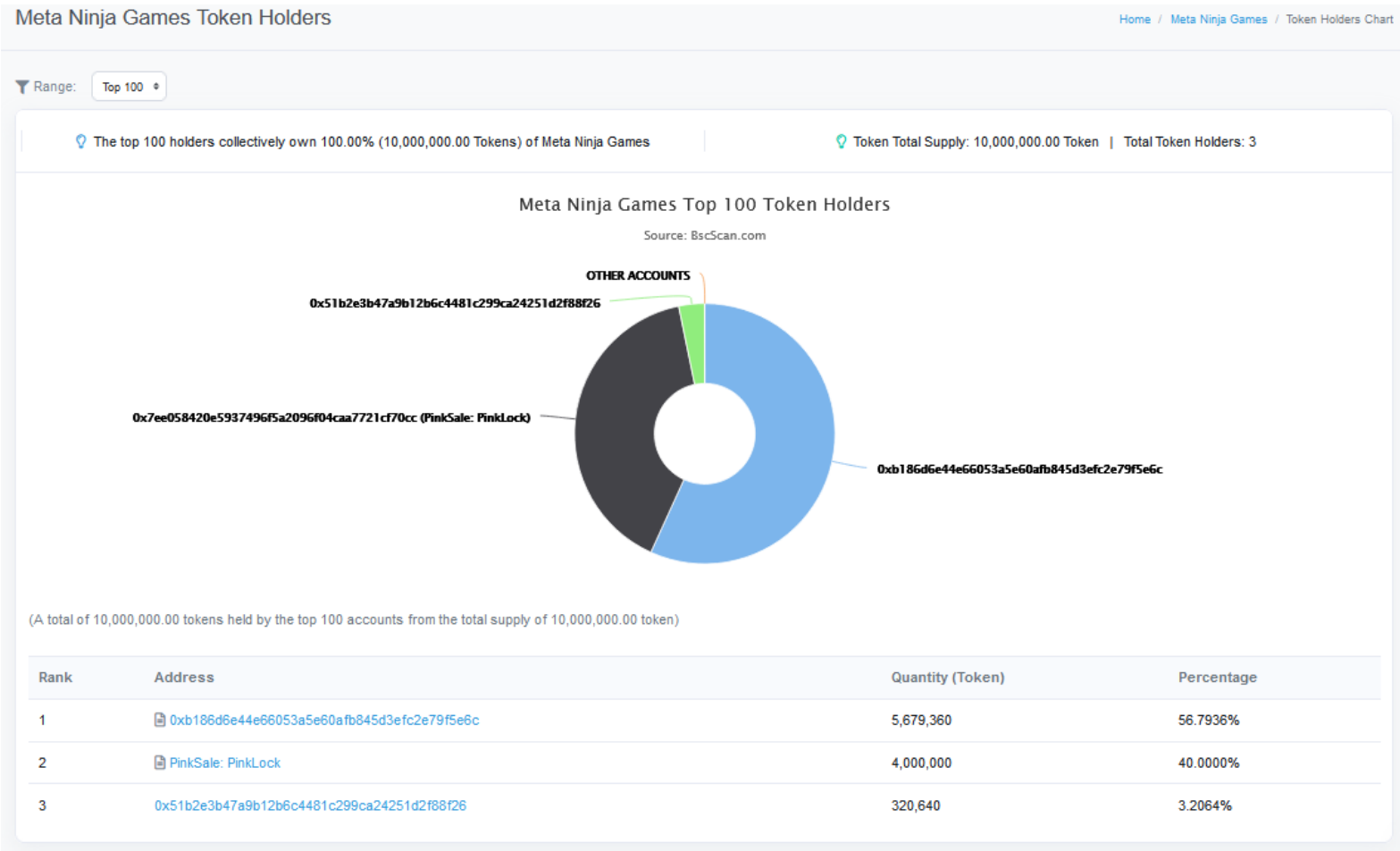
Analytix Audit in no way takes responsibility for any losses, nor does Analytix Audit encourage any speculative investments

VI.

## **INFORMATION**

The information provided in this audit is for information purposes only and should not be considered investment advice. Analytix Audit does not endorse, recommend, support, or suggest any projects that have been audited.

# TOKENOMICS - OVERVIEW



(This comment refers to when the audit was carried out maybe team locked tokens later)

# CONTRACT CODE - OVERVIEW



## QUICK REVIEW

- Owner can't change fees
- Owner can set tx Limits

(minium value 0.1% of supply)

# Owner Privileges:

```
520 * Requirements:
521 *
522 * - the calling contract must have an ETH balance of at least `value`.
523 * - the called Solidity function must be `payable`.
524 *
525 * _Available since v3.1._
526 */
527 function functionCallWithValue(
528     address target,
529     bytes memory data,
530     uint256 value
531 ) internal returns (bytes memory) {
532     return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
533 }
534
535 /**
536 * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256}[`functionCallWithValue`], but
537 * with `errorMessage` as a fallback revert reason when `target` reverts.
538 *
539 * _Available since v3.1._
540 */
541 function functionCallWithValue(
542     address target,
543     bytes memory data,
544     uint256 value,
545     string memory errorMessage
546 ) internal returns (bytes memory) {
547     require(address(this).balance >= value, "Address: insufficient balance for call");
548     require(isContract(target), "Address: call to non-contract");
549
550     (bool success, bytes memory returndata) = target.call(value: value)(data);
```

```
modifier onlyOwner() {
    function renounceOwnership() public virtual onlyOwner {
function transferOwnership(address newOwner) public virtual onlyOwner {
    function distributeETHDividends(uint256 amount) public onlyOwner {
    function updateDividendTracker(address newAddress) public onlyOwner {
function updateUniswapRouter(address newAddress) external onlyOwner {
function excludeFromFees(address account, bool excluded) public onlyOwner {
    function claimStuckTokens(address _token) external onlyOwner {
        function excludefromLimit(address account, bool excluded)
    function setBlackList(address addr, bool value) external onlyOwner {
    function excludeFromReward(address account) external onlyOwner {
        function setAutomatedMarketMakerPair(address pair, bool value)
function updateGasForProcessing(uint256 newValue) public onlyOwner {
    function updateClaimWait(uint256 claimWait) external onlyOwner {
    function setMarketingWallet(address newWallet) external onlyOwner {
        function setSwapEnabled(bool value) external onlyOwner {
        function setMaxWallet(uint256 value) external onlyOwner {
        function setMaxBuyAmount(uint256 value) external onlyOwner {
        function setMaxSellAmount(uint256 value) external onlyOwner {
function excludeFromDividends(address account) external onlyOwner {
function updateClaimWait(uint256 newClaimWait) external onlyOwner {
    function setBalance(address payable account, uint256 newBalance)
    function processAccount(address payable account, bool automatic)
```

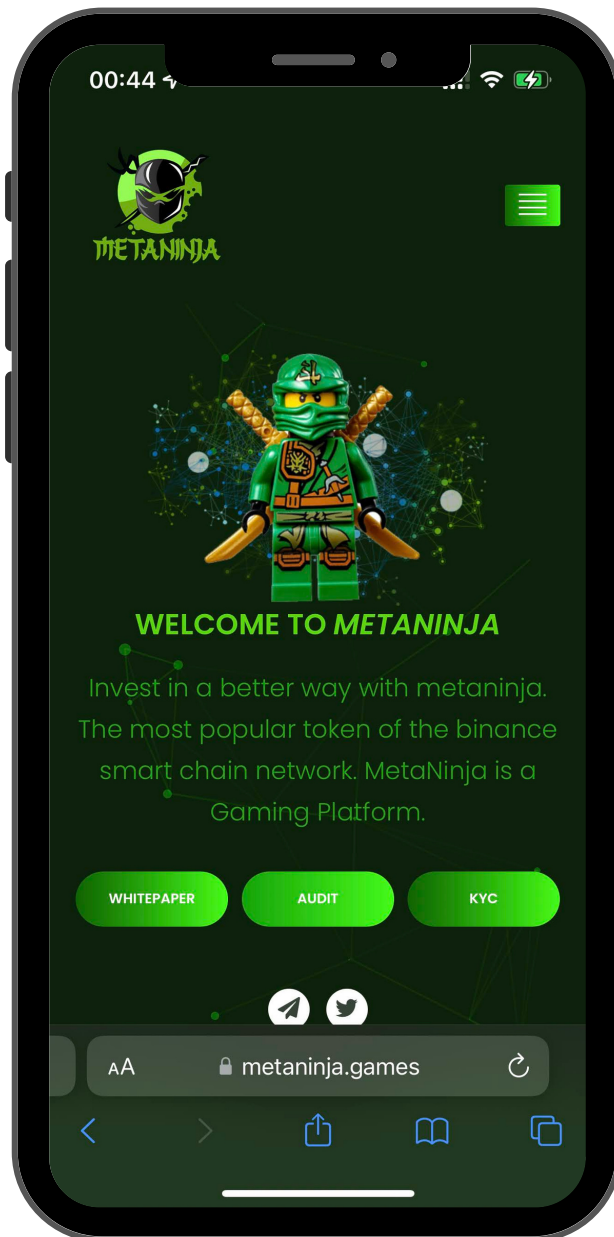
# Issue Checking:

N°	Issue description.	Checking Status.
1	Compiler Errors	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module	Passed
21	Fallback function security.	Passed

Audit Result:

**PASSED**

# WEBSITE - OVERVIEW



🕒 **Domain Registration: 2022-01-30**

33 days old

**SSL CERTIFICATE: A**

(<https://www.ssllabs.com>)

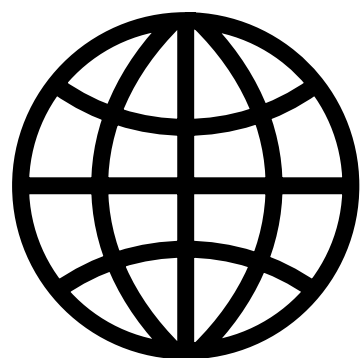
# SOCIAL MEDIA - OVERVIEW



**METANINJA\_OFFICIAL**



**METANINJAANN**



**WWW.METANINJA.GAMES/**